

CLAIMS

What is claimed is:

1 1. A method of allocating a processing resources to functions in a queue waiting to
2 be executed, comprising the steps of:
3 determining an amount of the processor resource available to be assigned;
4 determining an estimate of an amount of the resource needed for each function
5 waiting in the queue to execute; and
6 allocating the available resource to the functions based on a hierarchical priority
7 scheme.

1 2. The method of claim 1, wherein:
2 the functions are decomposed elements of a more complex process and do not
3 require the same amount of resource to execute.

1 3. The method of claim 2, wherein:
2 multiple instances of any function within the process may be invoked by the
3 processor to execute concurrently.

1 4. The method of claim 3, wherein:
2 each of the functions within the process is assigned a separate priority within the
3 hierarchical priority scheme.

1 5. The method of claim 4, wherein:
2 each instance of each function within the process is assigned a separate priority
3 within the hierarchical priority scheme.

1 6. The method of claim 2, further comprising the steps of:
2 assigning a resource throttling value to each function waiting in the queue to be
3 executed, wherein the throttling value determines the reduction of the resource allocated
4 to each of the functions.

1 7. The method of claim 1, wherein:
2 the allocation of the available resource to the functions waiting in the queue is
3 conducted to optimize the amount of the resource assigned to these functions.

1 8. The method of claim 1, wherein:
2 the allocation of the available resource to the functions waiting in the queue is
3 conducted to optimize a combined number of instances of each function concurrently
4 executed.

1 9. The method of claim 1, further comprising the steps of:
2 measuring the actual amount of the resource used;
3 revising the estimate of the amount of the resource needed for each function
4 waiting in the queue to execute based on the measured amount of the resource used; and
5 reallocating the available amount of the resource to the functions in accordance
6 with the revised estimate and the hierarchical priority scheme.

1 10. The method of claim 9, further comprising the steps of:
2 comparing the measured amount of the resource used to a high and a low threshold
3 value;
4 setting an alarm if the measured amount of the resource used exceeds the high
5 threshold value; and
6 removing the alarm if the measured amount of the resource used is less than the

7 low threshold value.

1 11. The method of claim 10, further comprising the step of:
2 assigning a resource throttling value to each function waiting in the queue to be
3 executed when the alarm is set, wherein the throttling value determines the reduction of
4 the resource allocated to each function.

1 12. The method of claim 10, further comprising the step of:
2 reducing the number of instances in which a particular function may execute
3 concurrently when the alarm is set.

1 13. The method of claim 5, further comprising the steps of:
2 measuring the actual amount of the resource used;
3 revising the estimate of the amount of the resource needed for each function
4 waiting in the queue to execute based on the measured amount of the resource used; and
5 reallocating the available amount of the resource to the instances in accordance
6 with the revised estimate and the hierarchical priority scheme.

1 14. The method of claim 13, further comprising the steps of:
2 comparing the measured amount of the resource used to a high and a low threshold
3 value;
4 setting an alarm if the measured amount of the resource used exceeds the high
5 threshold value; and
6 removing the alarm if the measured amount of the resource used is less than the
7 low threshold value.

1 15. The method of claim 14, further comprising the step of:

2 assigning a resource throttling value to each instance of each function waiting in
3 the queue to be executed when the alarm is set, wherein the throttling value determines
4 the reduction of the resource allocated to each instance of each of the functions.

1 16. The method of claim 14, further comprising the step of:

2 reducing the number of instances in which a particular function may execute
3 concurrently when the alarm is set.

17. A method of allocating a processing resources to functions in a queue waiting to
be executed, comprising the steps of:

determining an amount of the processor resource available to be assigned;

for each of j instances of k functions, calculating a product obtained by:

- (a) estimating the amount of resource needed to support the execution of
the j^{th} instance of the k^{th} function;
- (b) assigning a value of either zero or one to a multiplicand associated
with the j^{th} instance of the k^{th} function; and
- (c) multiplying the estimated amount of resource needed to support the
execution of the j^{th} instance of the k^{th} function by its associated
multiplicand and assigning the result to the product associated with
the j^{th} instance of the k^{th} function;

for each of the j instances, calculating a sub-total sum obtained by:

- (d) summing together the products associated with each of the k
functions of the j^{th} instance; and
- (e) adding an estimate of the resource needed to support background
processing associated with the j^{th} instance to the sum of the products
associated with each of the k functions of the j^{th} instance and
assigning the result to the sub-total for the j^{th} instance; and

allocating the available resource to the k functions of the j instances based on a hierarchical priority scheme.

1 18. The method of claim 17, wherein:

2 the multiplicand value associated with the j^{th} instance of the k^{th} function is
3 determined according to the hierarchical priority scheme.

1 19. The method of claim 17, further comprising the step of:

2 repeating the steps recited in claim 17 for each of a number of sequential time
3 periods.

1 20. The method of claim 19, wherein:

2 the length of each time period is variable and is no longer than the period needed
3 to execute any one of the j instances of the k functions that are executing concurrently.

1 21. The method of claim 19, further comprising the step of:

2 for each of the j instances of the k^{th} function, assigning increasingly higher priority
3 in accordance with an increasingly greater number of time periods that have passed since
4 the j^{th} instance of the k^{th} function was last executed.

1 22. The method of claim 17, further comprising the steps of:

2 measuring the actual amount of the resource used;
3 revising the estimate of the amount of the resource needed for each function
4 waiting in the queue to execute based on the measured amount of the resource used; and
5 reallocating the available amount of the resource to the instances of each function
6 in accordance with the revised estimate and the hierarchical priority scheme.

1 23. The method of claim 22, further comprising the steps of:
2 comparing the measured amount of the resource used to a high and a low threshold
3 value;
4 setting an alarm if the measured amount of the resource used exceeds the high
5 threshold value; and
6 removing the alarm if the measured amount of the resource used is less than the
7 low threshold value.

1 24. The method of claim 23, further comprising the step of:
2 assigning a resource throttling value to each instance of each function waiting in
3 the queue to be executed when the alarm is set, wherein the throttling value determines
4 the reduction of the resource allocated to each instance of each of the functions.

1 25. The method of claim 23, further comprising the step of:
2 reducing the number of instances in which a particular function may execute
3 concurrently when the alarm is set.